



# EKS を利用した決済基盤での DevOpsへの取り組み

2022.08.04

夏のAWS Kubernetes祭り! ~Kubernetesの運用最前線を紹介~

# 上妻 靖広

# 西村 真一

## OpenSaaS Studio / Payment 所属

- **2014年 サイバーエージェントに入社**

ゲーム、アドテクなど様々なプロダクトの  
バックエンドエンジニアとして従事

- **2020年 OpenSaaS Studio に所属**

当初は SRE 的ポジションとしてデプロイ効率化に取り組み、現在は SDK の開発を行っている

## OpenSaaS Studio / Payment 所属

- **2013年 サイバーエージェントに入社**

様々なスマホアプリのバックエンドを開発  
2年ほどAndroidエンジニアに転向

- **2018年 OpenSaaS Studio に所属**

現在は SRE 的ポジションとして決済基盤の信頼性  
向上、セキュリティ向上に取り組んでいる

# Contents

1. Simply について
2. 構成について
3. 運用について
4. 改善したい点
5. まとめ



1

# Simply について



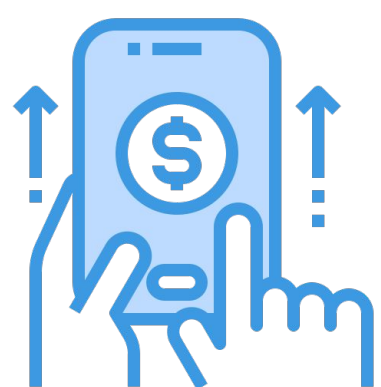
# Simplyについて

---



## Simply サービス内容

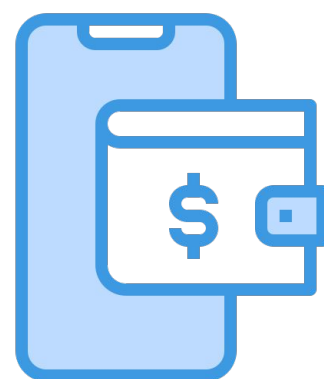
サイバーエージェントの社内向けの決済基盤の1つ



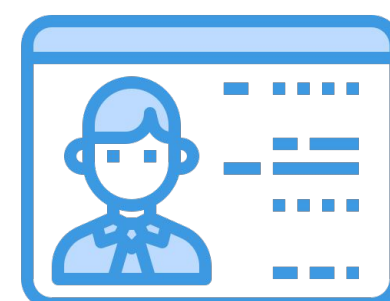
ペイメント



サブスクリプション



ウォレット



本人確認

# Simplyについて

## チーム構成

開発 or SRE



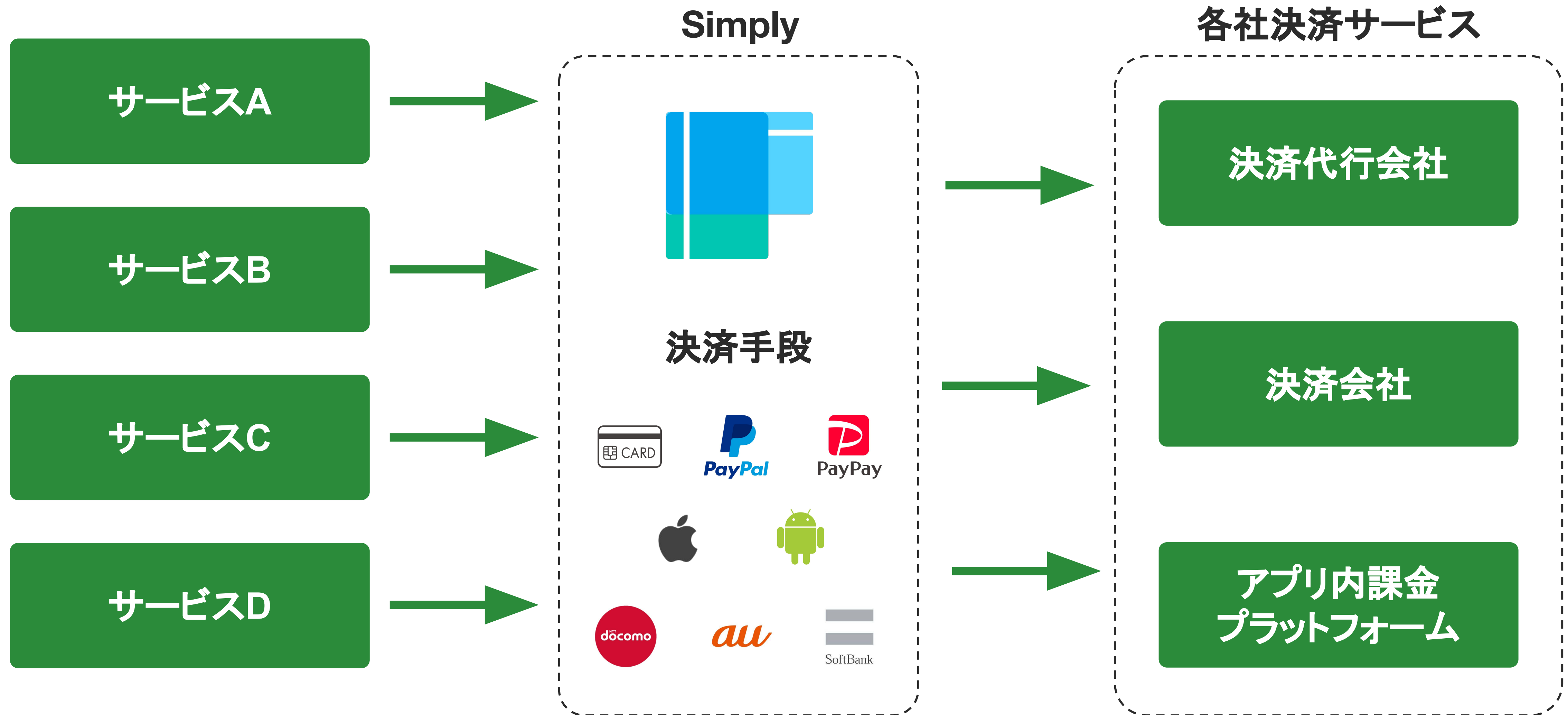
プロダクト  
マネージャー



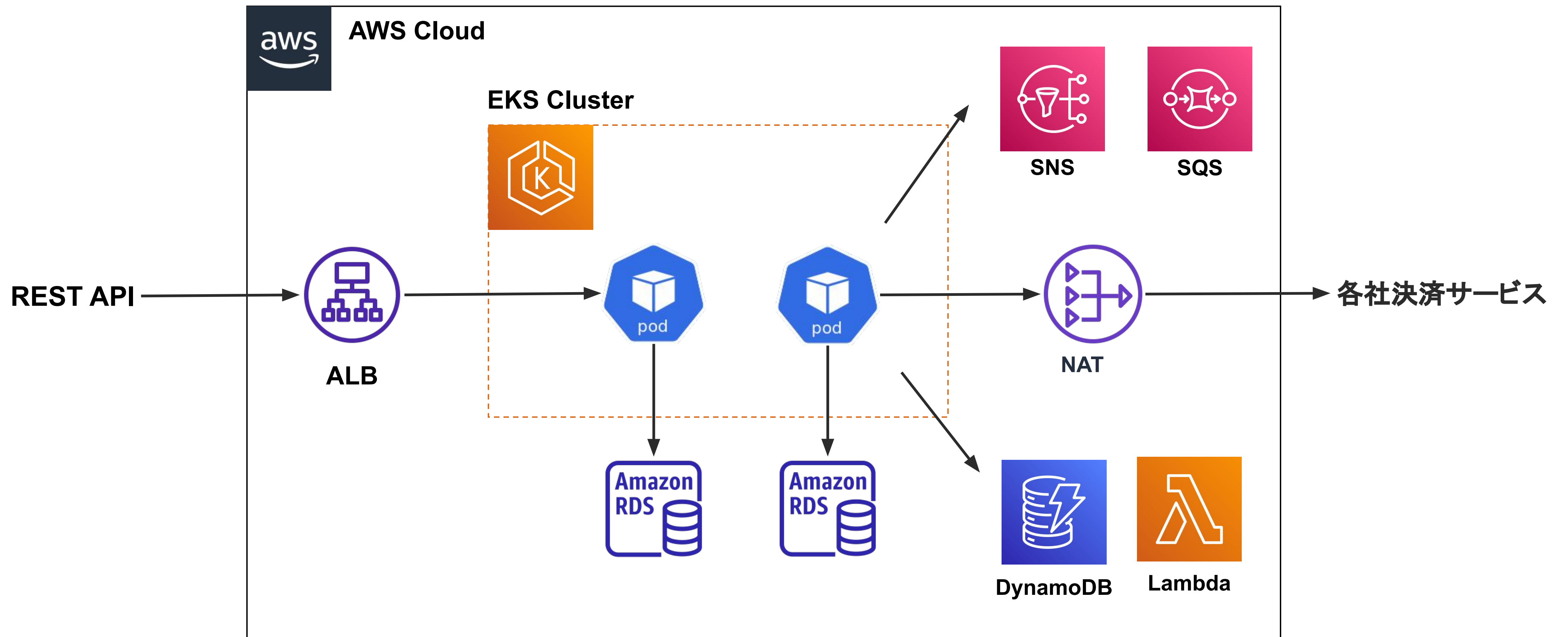
プロジェクト  
マネージャー

- 全員ソフトウェアエンジニア
- 状況に合わせて役割を変更
- 専任のリリース担当者はいない
- 基本的には、チームメンバーだけで開発からインフラ運用まで

# 全体図



# アーキテクチャ図



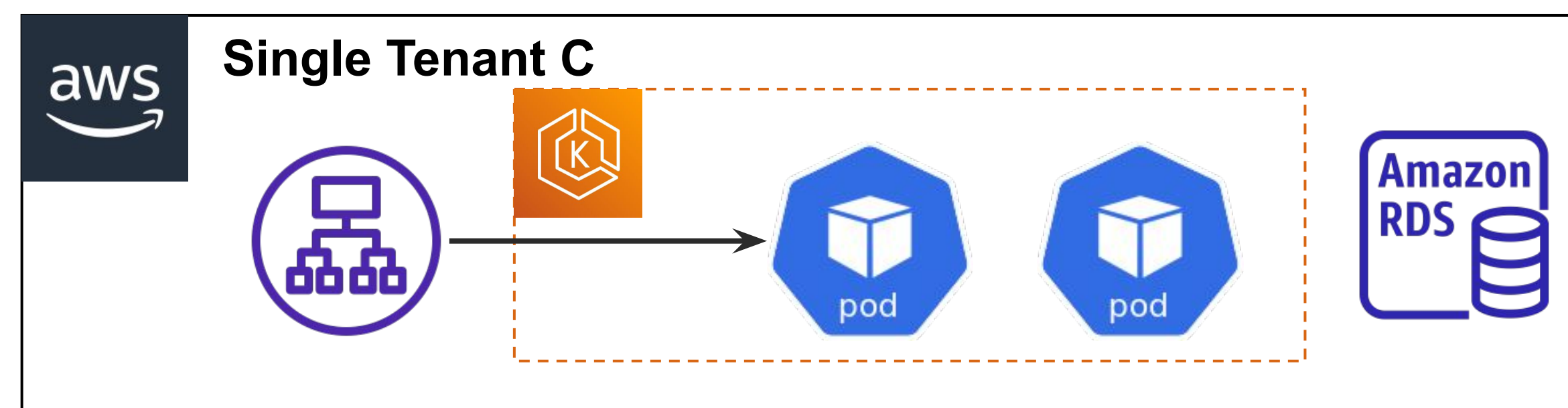
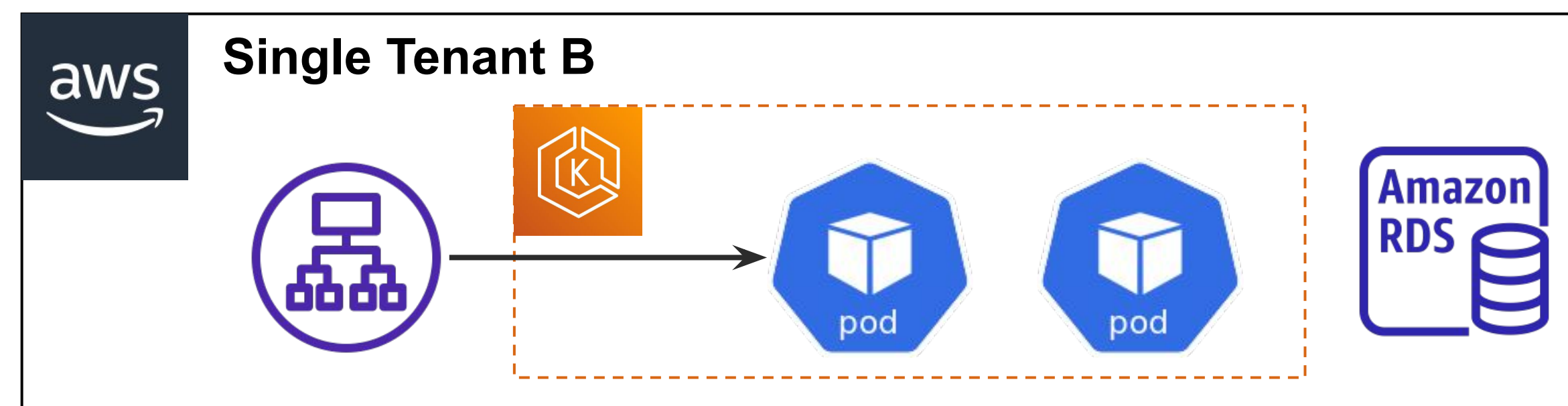
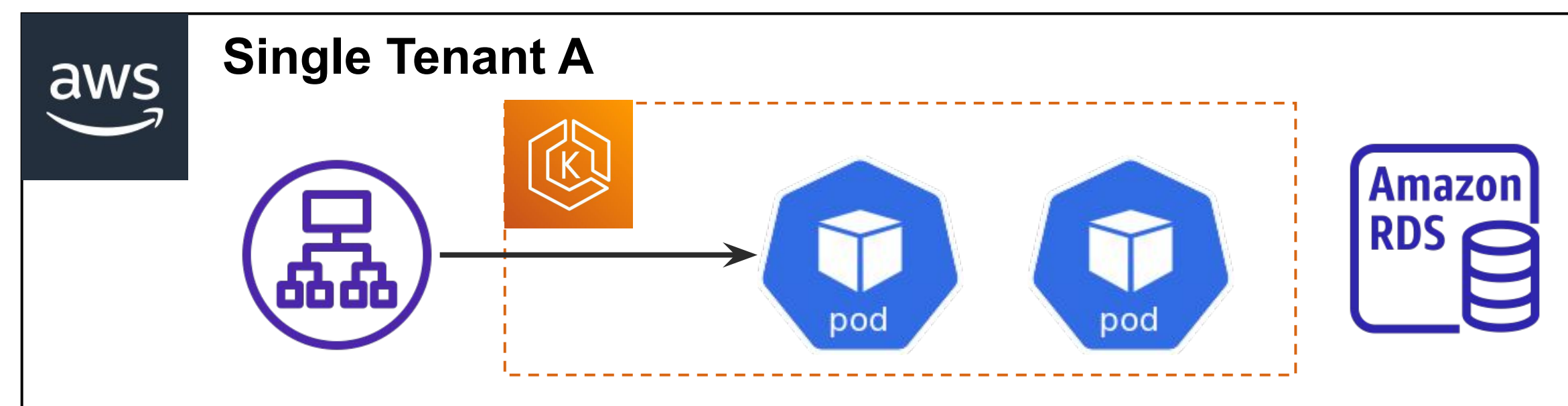
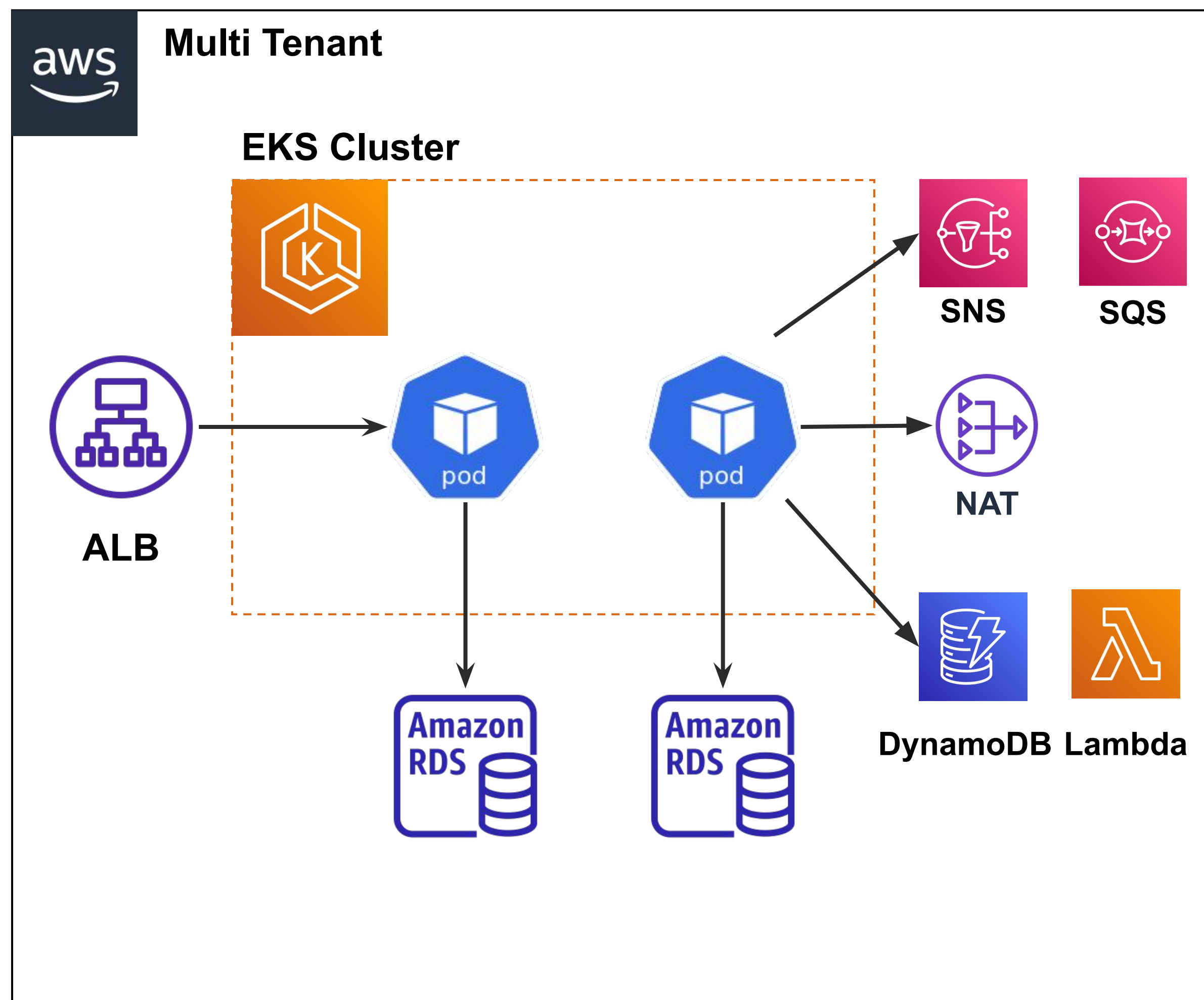


2

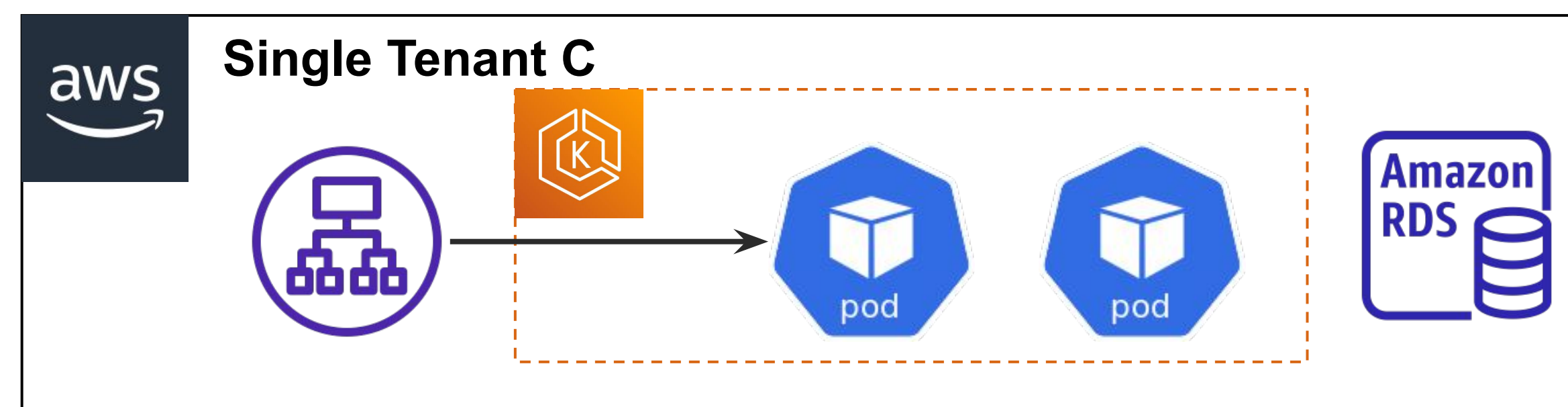
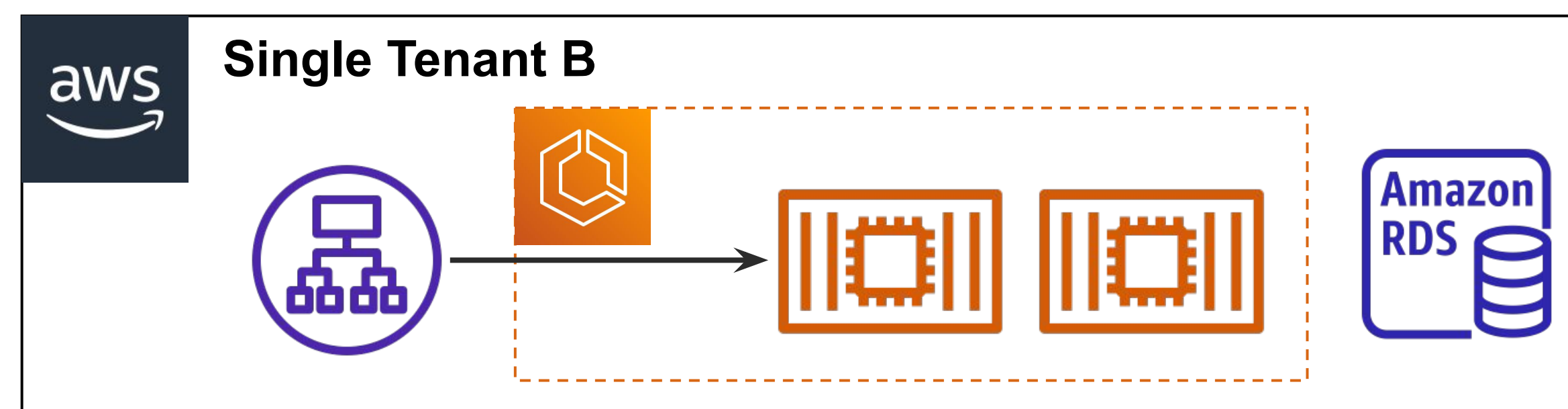
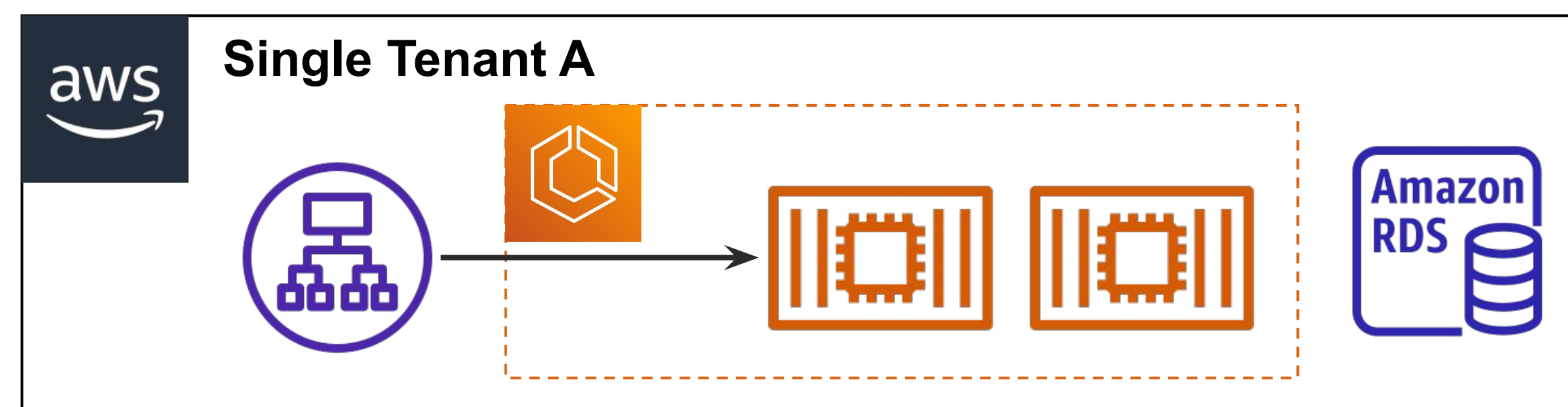
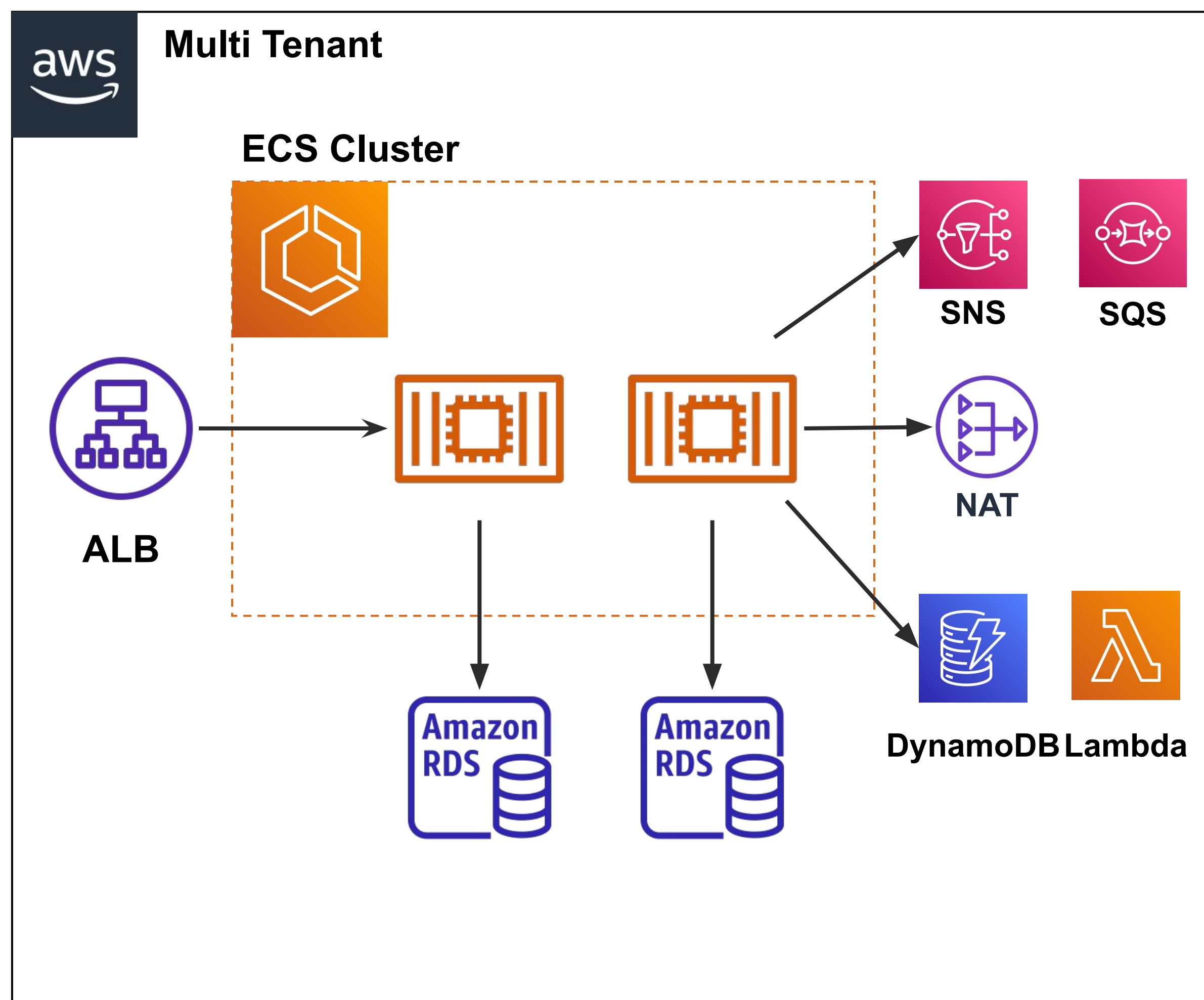
# 構成について



# システム構成図



# システム構成図(1年前)



# EKS への統一

---

## 目的

- **運用コストの低減**  
1つのモジュールを全環境にデプロイするだけでまる1日程度かかることがあった
- **アーキテクチャの統一**  
いろんなサービスを利用しているので、運用時や障害対応時等に複雑になっていた

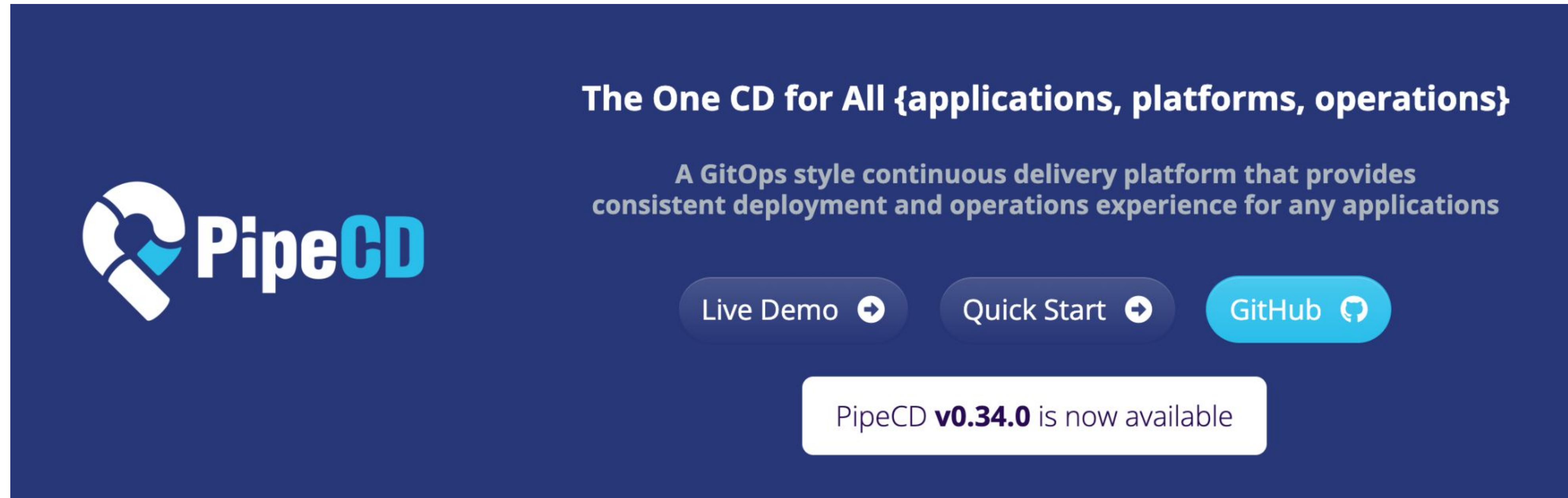
## 実施結果(Pros/Cons)

- **運用コストの低減**  
GitOps を意識したデプロイフローにすることによって、大幅な改善がみられた
- **Kubernetes のバージョンへの追従**  
定期的にサポート切れになるKubernetes のバージョンへの追従を強いられ得ることとなった



# PipeCD について

---

A promotional banner for PipeCD on a dark blue background. On the left is the PipeCD logo, which consists of a stylized white and blue icon followed by the text "PipeCD" in white and blue. To the right of the logo, the text "The One CD for All {applications, platforms, operations}" is written in white. Below this, a smaller line of text reads "A GitOps style continuous delivery platform that provides consistent deployment and operations experience for any applications". At the bottom of the banner are three buttons: "Live Demo" with a right-pointing arrow, "Quick Start" with a right-pointing arrow, and "GitHub" with the GitHub logo. Below these buttons is a white box containing the text "PipeCD v0.34.0 is now available".

The One CD for All {applications, platforms, operations}

A GitOps style continuous delivery platform that provides consistent deployment and operations experience for any applications

Live Demo → Quick Start → GitHub

PipeCD v0.34.0 is now available

<https://pipecd.dev/>

## GitOps スタイルの CD ツール

様々な種類のアプリケーション(Kubernetes, ECS, Terraform等)を  
様々なプラットフォーム(AWS, GCP等)にデプロイ可能

# PipeCD について

The screenshot displays the PipeCD dashboard for an application named 'payment-api'. The top navigation bar includes 'Applications', 'Deployments', and 'Chains'. A deployment status is shown as 'SUCCESS 6 minutes ago' for the environment 'shared-prd'. Below this, a message states 'The deployment was completed successfully'. Metadata includes the application name 'payment-api', the pipeline 'shared-prd', and the commit 'Merge pull request #2342 from simply-app/feature/prd-release (c48cd97)'. The deployment was triggered by 'kozuma-yasuhiro'. A summary indicates 'Sync progressively because of updating image'. A pipeline diagram shows four steps: 'K8S\_CANARY\_ROLLOUT', 'WAIT\_APPROVAL' (with a note 'Approved by kozuma-yasuhiro'), 'K8S\_PRIMARY\_ROLLOUT', and 'K8S\_CANARY\_CLEAN', all of which are marked as successful.

## 管理対象

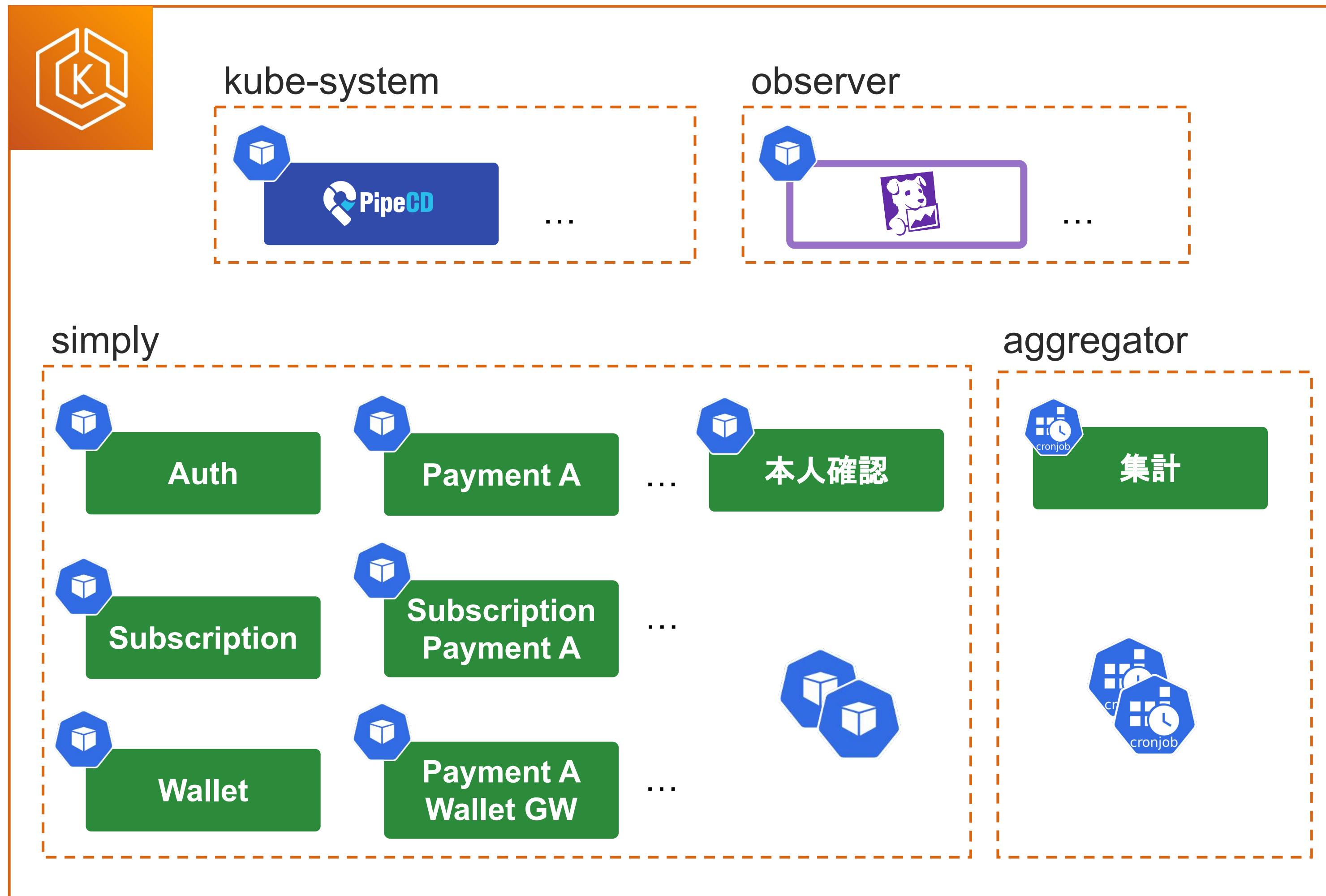
EKS のすべての Deployment, CronJob を管理している

## デプロイ方法

Github の定義が更新されると自動的にデプロイが開始され、パイプラインにてデプロイされる

# EKS クラスタについて

EKS Cluster



- Namespace は細かく分けていない
- 決済手段や役割ごとに Deploymentを分割
- Deployments **50+**
- Pods **100+**

# EKS クラスタについて

## 環境ごとの違い

	Payments	Subscription Payments	Wallet	本人確認
Multi Tenant	14	8	○	○
Single Tenant A	4	7	○	-
Single Tenant B	5	0	-	○
Single Tenant C	6	6	○	-

- **Multi Tenant 環境**  
全機能を提供
- **Single Tenant 環境**  
必要な機能に絞って提供



# EKS クラスタについて

---

## アドオンについて

### EKS デフォルト

amazon-vpc-cni, coredns, kube-proxy

### メトリクス, モニタリング系

cluster-autoscaler

metrics-server

datadog-agent

datadog-cluster-agent

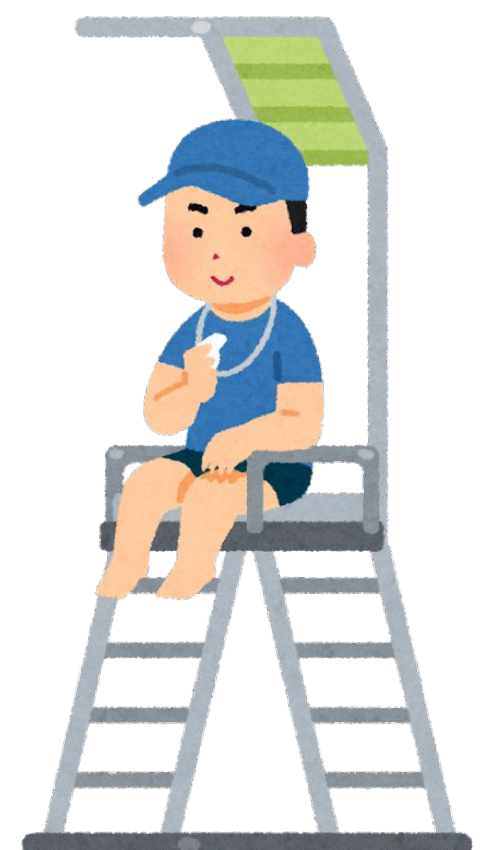
### 管理系

aws-load-balancer-controller

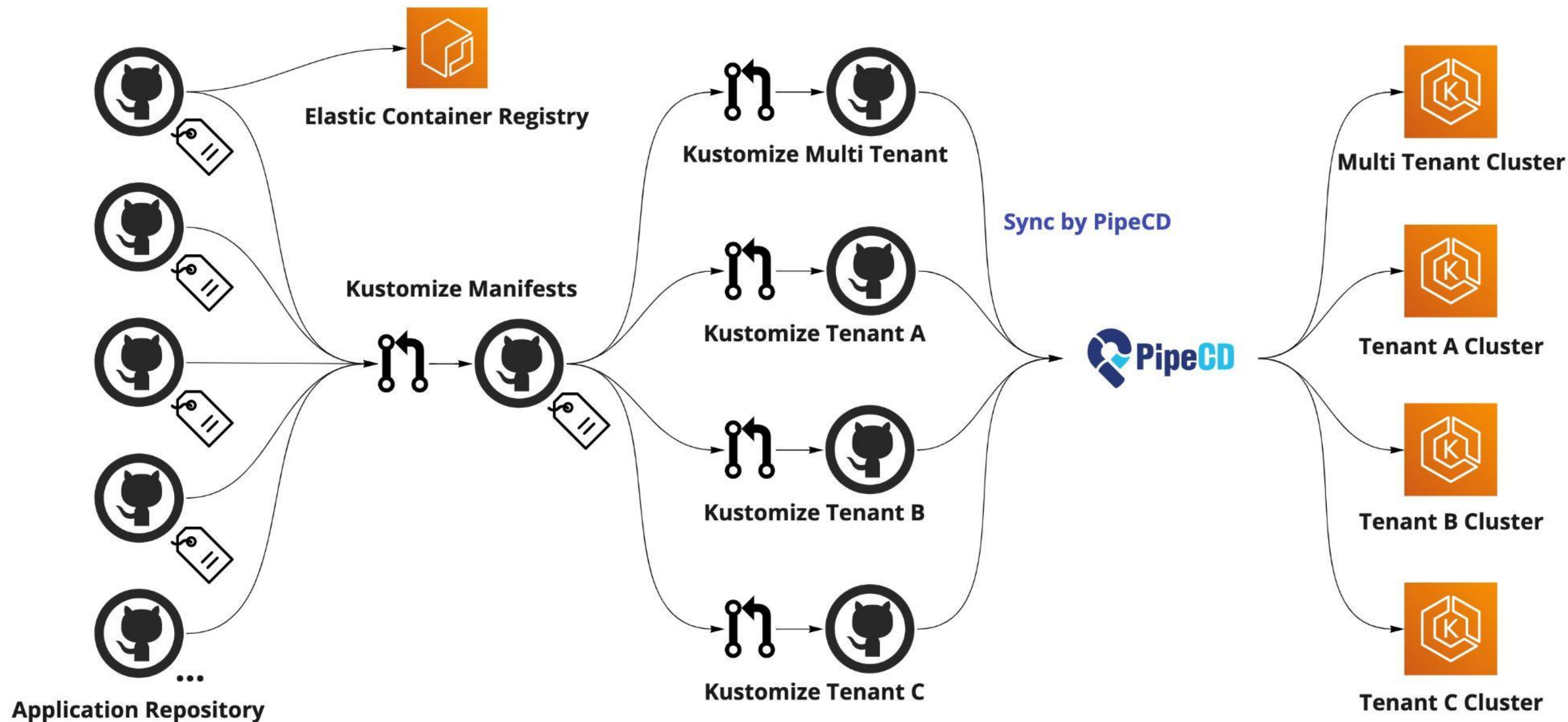
cert-manager

secret-manager

必要最低限のアドオンを使っている

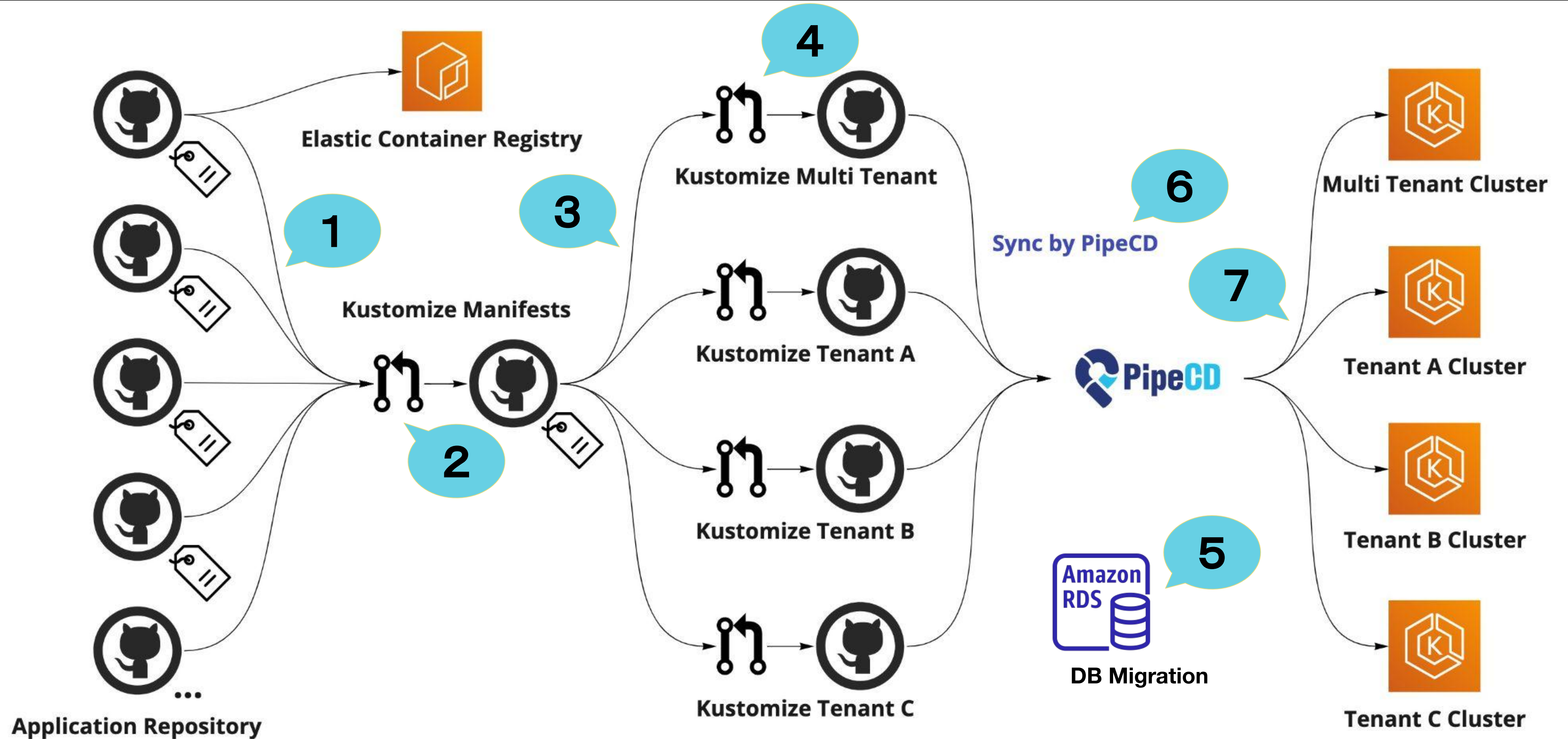


# デプロイフローについて





# デプロイフローについて



# 監視について

---

## Datadog によるモニタリング

- 決済取引の監視

利用者からみた直接的な障害の検知が目的

- ・決済失敗数、エラー数、Latency など

- EKS の監視

障害の予兆を捉えることが目的

- ・重要なリソースの空き容量
- ・異常な動きの検知

... 詳しくは次のページで



# 監視について

---

## EKS の監視

- **重要なリソースの空き容量の監視**

- ・Node のオートスケール上限
- ・HPA で割当可能な Pod 数
- ・サブネットを利用可能な IP 数

...

- **異常な動きの検知**

- ・SCHEDULABLE ではない Node 数
- ・正常に配置できていない DaemonSet
- ・再起動を繰り返している Pod

...

**EKS 関連のモニター数: 18**

**EKS 関連のアラートはあまり発生していない**



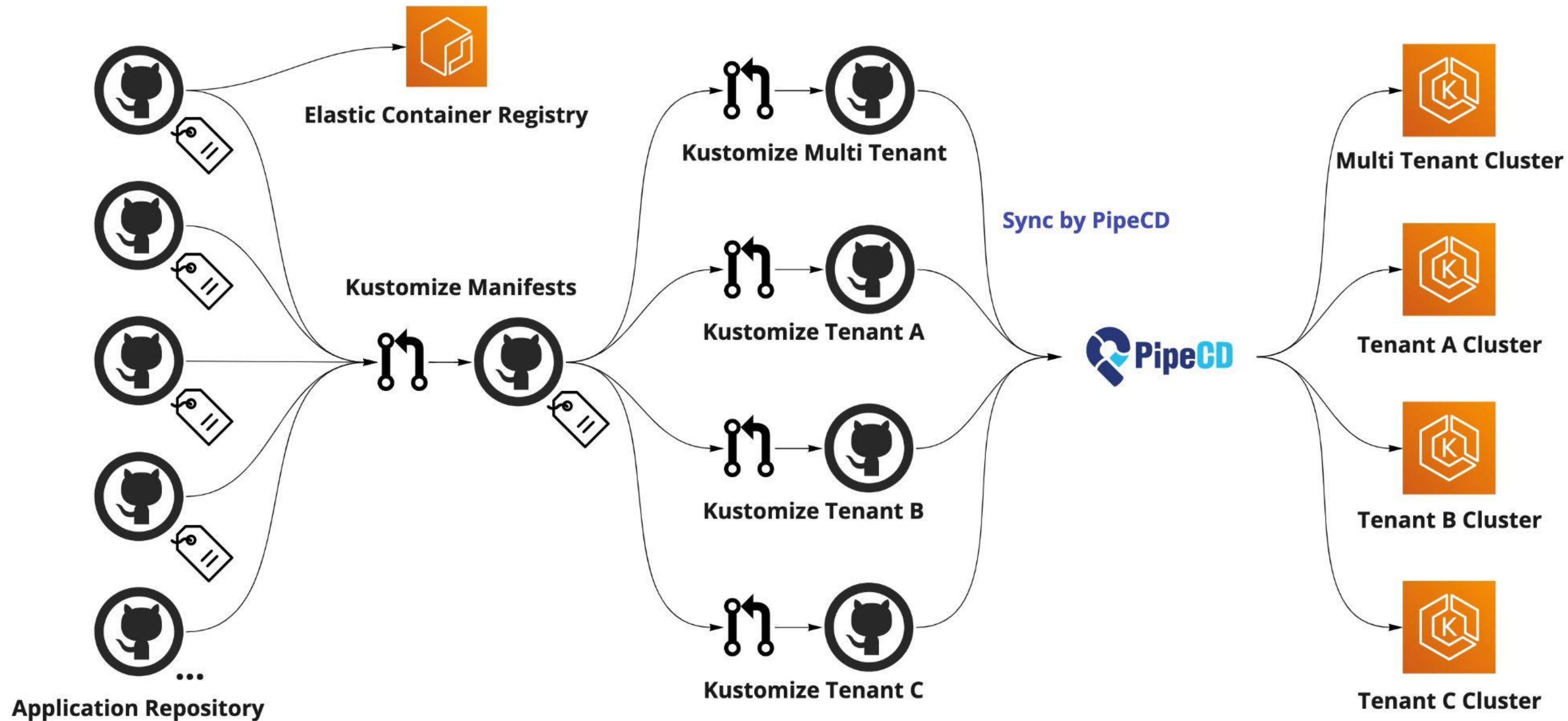
3

運用について





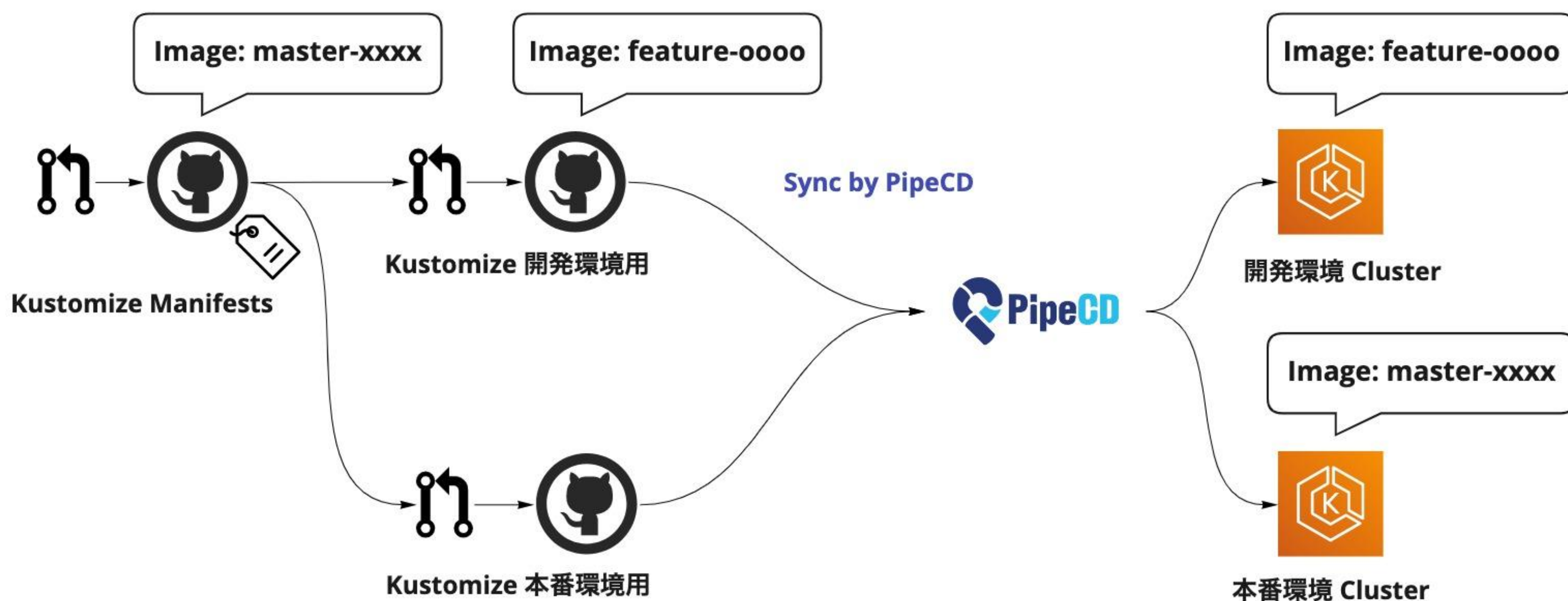
# デプロイ戦略



# デプロイ戦略

## Image を固定したい場合

開発中のモジュールを開発環境にデプロイしたい場合に利用しています



Kustomize を採用しているので、Kustomize Manifests で定義された Image を個別のリソース管理の方で上書きしています

リリース時に戻し忘れが発生しやすいのが玉にキズです



# デプロイ戦略

---

## カナリアデプロイについて

- PipeCD の機能を使ってカナリアデプロイを実践している
- 複数台の Pod で運用しているモジュールに対してデプロイ時に先行して 1 台だけカナリアデプロイを実施している

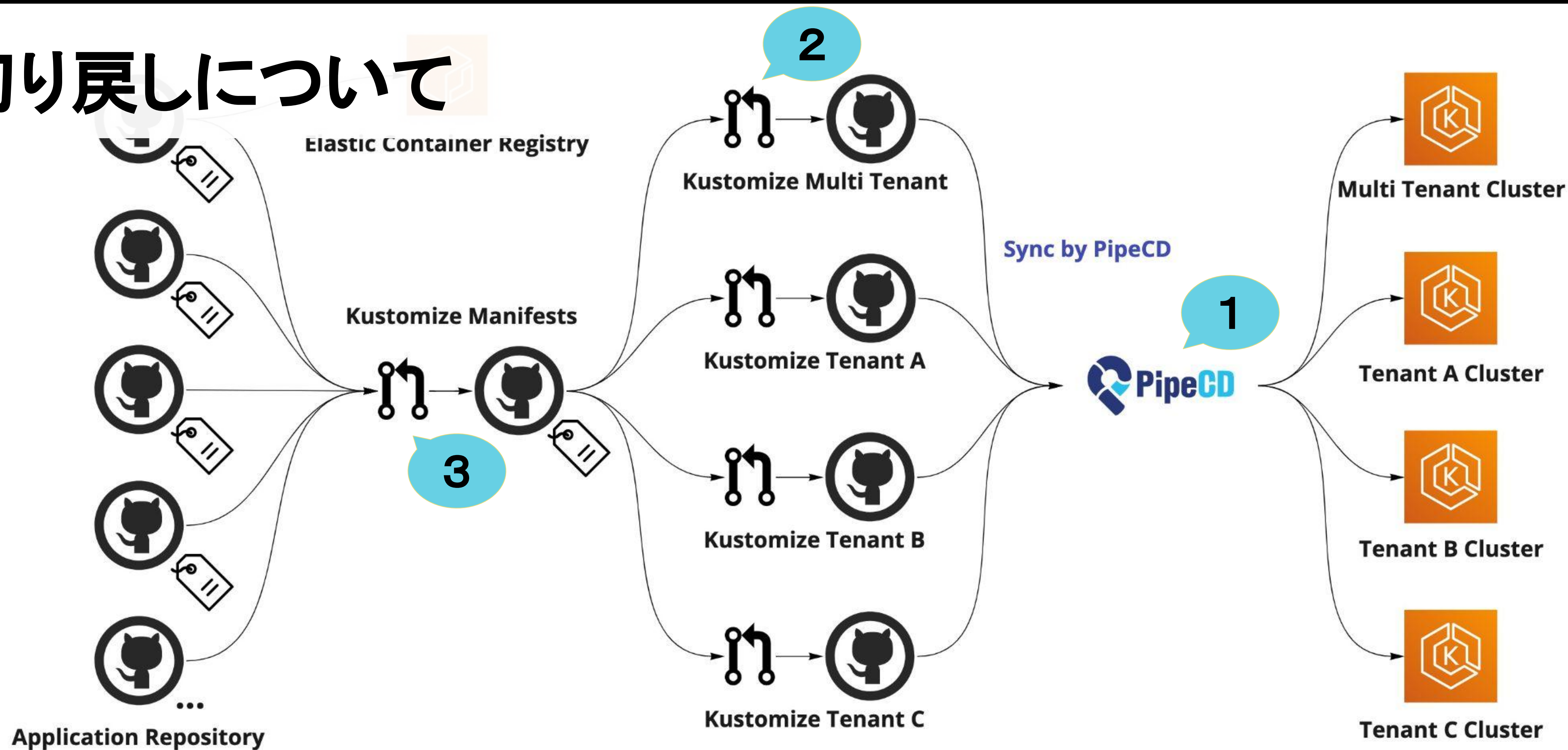
PipeCD的にはパーセンテージでも対応は可能だが、デフォルトの Pod のレプリカ数が最低 2 台とか少ないので固定で 1 台にしている

- デプロイした結果、問題がなければ手動承認でリリースを進めている

自動承認でのリリースも可能だが、リクエスト数が極端に少ない場合があるので判断が難しい

# デプロイ戦略

## 切り戻しについて



# Kubernetes のアップデート戦略

---

## 方針

- インプレースアップグレード  
サービスはメンテナンスに入れずに稼働したままEKSクラスタをアップデート
- 3ヶ月間隔でアップデート  
バージョンのサポートが切れる1ヶ月前にはアップデート完了するようにスケジュール
- クラスタの構成はシンプルに  
環境ごとの差異が最小限になるように構成をあわせる  
サードパーティのアドオンは極力使わない

# Kubernetes のアップデート戦略

## 手順



知識が偏らないように、かつ安全にアップデートする工夫

- ・調査はメンバーが持ち回りで実施。調査手順、アップデート手順は定形化
- ・アップデート作業はメンバーで分担。ペアオペでドライバーを交代しながら実施
- ・1環境ずつ振り返りを実施して、手順書に対してフィードバックを行う





# アドオンのアップデート戦略

---

## 方針

- 軽微なアップデートはKubernetesと同時にアップデート  
マイナーバージョンの変更など
- 大きなアップデートは事前にアップデート  
API Version の変更など  
aws-alb-ingress-controller → aws-load-balancer-controller
- 1環境単位ではなく、1アドオン単位で全環境アップデート  
全環境同じバージョンに揃える (GitOps 的にも自然とそうなる)

4

改善したい点



# 改善したい点

---

## デプロイフローの改善

- ・デプロイフローの自動化
  - ・End-to-End テスト
  - ・リリースの切り戻し
  - ・DB Migration
- ・デプロイ状況の可視化
- ・ビルド並列化によるリードタイム短縮

## Kubernetes環境の改善

- ・Kubernetes アップデートのスケジュールの前倒し
- ・Kubernetes やアドオンの新しい機能の有効活用



5

まとめ





# まとめ

---

- **アーキテクチャを統一したことによって、複数のAWSアカウントへのデプロイが簡単になった**
- **GitOps を導入することによって、リリース作業が大幅に短縮された**
- **PipeCD を利用することによって、デプロイの自動化が進んだ**
- **Kubernetes の運用は大変**  
Kubernetes のバージョンのアップデートへの追従、アドオンの管理
- **デプロイフローの改善は継続していきたい**  
各種自動化、デプロイ状況の可視化



# 一緒に働きませんか？

---

**随時募集中です！**

採用情報 | 株式会社サイバーエージェント

<https://www.cyberagent.co.jp/careers/>



ありがとうございました